

**Mand2000Rexx**

**COLLABORATORS**

	<i>TITLE :</i> Mand2000Rexx		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 17, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Mand2000Rexx</b>	<b>1</b>
1.1	Mand2000Rexx.guide . . . . .	1
1.2	installation . . . . .	2
1.3	syntax_overview . . . . .	2
1.4	misc . . . . .	2
1.5	printing_from_arexx . . . . .	3
1.6	startupscript . . . . .	3
1.7	portnames . . . . .	4
1.8	commands . . . . .	4
1.9	AppName/OpenAnim . . . . .	9
1.10	AppName/OpenScreenAnim . . . . .	9
1.11	AppName/WriteAnimFrame . . . . .	9
1.12	AppName/CloseAnim . . . . .	10
1.13	AppName/MinimizeHammies . . . . .	10
1.14	AppName/Help (Global) . . . . .	10
1.15	AppName/GETATTR (Global) . . . . .	11
1.16	AppName/Help (Project) . . . . .	11
1.17	AppName/Verbose . . . . .	12
1.18	AppName/Menu . . . . .	12
1.19	AppName/ClearMenus . . . . .	13
1.20	AppName/ClearNMenus . . . . .	13
1.21	AppName/EventAction . . . . .	13
1.22	AppName/LockGui . . . . .	14
1.23	AppName/UnlockGUI . . . . .	15
1.24	AppName/IncPause . . . . .	15
1.25	AppName/DecPause . . . . .	15
1.26	AppName/NOP . . . . .	16
1.27	AppName/SetMorphFactor . . . . .	16
1.28	AppName/SetExtraAcc . . . . .	16
1.29	AppName/Scroll . . . . .	16

---

---

1.30	AppName/GetAttr (Project)	16
1.31	AppName/New	18
1.32	AppName/NewJulia	18
1.33	AppName/Open	18
1.34	AppName/Reset	19
1.35	AppName/Save	19
1.36	AppName/SaveAs	19
1.37	AppName/SaveScreenAs	19
1.38	AppName/Close	20
1.39	AppName/Print	20
1.40	AppName/Quit	20
1.41	AppName/ZoomInToPixel	20
1.42	AppName/ZoomOutToPixel	21
1.43	AppName/ZoomInToFloat	21
1.44	AppName/ZoomOutToFloat	22
1.45	AppName/SetLocation	22
1.46	AppName/SetJulia	22
1.47	AppName/SetFractalType	23
1.48	AppName/GotoGood	23
1.49	AppName/Undo	23
1.50	AppName/Redo	24
1.51	AppName/ForceRedraw	24
1.52	AppName/ForceRecalc	24
1.53	AppName/CycleColours	24
1.54	AppName/OpenColours	25
1.55	AppName/SetColour	25
1.56	AppName/SetColourMapping	25
1.57	AppName/SetColourMap	26
1.58	AppName/SetScreen	26
1.59	AppName/MaxIters	26
1.60	AppName/NumDrawPasses	27
1.61	AppName/TitleBar	27
1.62	AppName/BackDrop	27
1.63	AppName/Workbench	27
1.64	AppName/FreeBuffers	28
1.65	AppName/MandToFront	28
1.66	AppName/MandToBack	28
1.67	AppName/MoveWindow	28
1.68	AppName/SizeWindow	29

---

---

1.69	AppName/ChangeWindow . . . . .	29
1.70	AppName/WindowToFront . . . . .	29
1.71	AppName/WindowToBack . . . . .	29
1.72	AppName/ActivateWindow . . . . .	30
1.73	AppName/ZoomWindow . . . . .	30
1.74	AppName/RequestResponse . . . . .	30
1.75	AppName/RequestNotify . . . . .	30
1.76	AppName/RequestFile . . . . .	31
1.77	AppName/RequestSaveFile . . . . .	31
1.78	AppName/RequestString . . . . .	31
1.79	AppName/Requester . . . . .	32
1.80	AppName/DisplayMessage . . . . .	32
1.81	AppName/OkayN . . . . .	33

---

# Chapter 1

## Mand2000Rexx

### 1.1 Mand2000Rexx.guide

Mand2000 ARexx Documentation - Copyright © 1993,1994 Cygnus ↵  
Software.  
Please do not distribute.

Mand2000 is a very complete program in itself, but no program can ever anticipate every possible use that people will have for it, which is when an ARexx port is invaluable. Through Mand2000's ARexx port you can create animations, setup custom colour mapping, add keyboard shortcuts to frequently used commands, and many more things.

Even if you don't know anything about ARexx, you should still install the ARexx scripts supplied with Mand2000, because they add some new menu commands to Mand2000. See below for installation instructions.

This document attempts to explain what the available ARexx commands for Mand2000 are, and how to use them. It does not attempt to teach ARexx. However ARexx is a fairly simple language, and there are example scripts provided, so with a bit of study and the ARexx manual you should be able to pick it up. Even if you decide not to write ARexx scripts, the scripts that are provided with Mand2000 are still quite useful, and other scripts should become available as they are written by other people.

#### Installation

Installing Mand2000's ARexx scripts

#### Syntax Overview

Overview of syntax

#### Misc

Misc stuff

#### Debugging

Printing to a console window from Mand2000

#### Startup script

Mand2000's startup ARexx script

---

Port names  
Mand2000's ARexx port names

Commands  
ARexx command descriptions

## 1.2 installation

### Installation

To install the ARexx scripts, go into the ARexx drawer on the Mand2000 distribution disk and double click on the 'InstallRexx' icon. This will copy the scripts to rexx:, where they will automatically be noticed by Mand2000 the next time you run it. However, in order for these scripts to show up and be used, ARexx must be running. The program RexxMast, which comes with workbench 2.04 and above, must be run, either from the workbench or the CLI. It can be found in the system drawer of your workbench disk. If you want it automatically run each time you turn on your computer, move it into the WBStartup drawer.

## 1.3 syntax\_overview

### Syntax Overview

The syntaxes for all of the commands are given in the standard AmigaDOS format. In this format a series of keywords, with flags that govern their behaviour, describe the syntax. Multiple flags can appear with a single parameter. These are all of the flags that are used in Mand2000:

/S - Switch. This parameter is true if the name appears, false otherwise.

/K - Keyword. This means that the keyword must be used if you want to specify this parameter.

/N - Number. A number is required for this parameter.

/A - Required. This means that this parameter is mandatory.

/F - Rest of line. Everything beyond this point is part of the parameter.

All commands are case insensitive.

Careful quoting will sometimes be needed in order to preserve spaces and case in some of your parameters. This is because ARexx strips off one layer of quotes and Mand2000 strips off another layer. If you start your strings with a single quote followed by a double quote, and end them with a double quote followed by a single quote, that should work.

## 1.4 misc

---

## Miscellaneous

When an ARexx script is called from Mand2000, the active window is paused by default. This is done to avoid strange synchronization problems where Mand2000 is fighting your script. This means that Mand2000 will stop calculating the active window when an ARexx script is executed, and will resume calculating when the script returns. To change this action, see the

DecPause  
and  
IncPause  
commands.

When an ARexx command encounters a serious error, this error is reported by setting the RC or return code variable to a non-zero value which roughly indicates the type of error. Since it is not possible to return any extra information in the result variable when an error occurs, extra information is put into MAND2000.LASTEROR. 'say' this variable if you are having trouble figuring out why a command is failing.

When errors do occur in Mand2000 ARexx scripts, the return codes put into the RC variable follow a consistent pattern, as indicated below:

- 3 - Warning - returned if the user cancels a requester.
- 4 - Disabled - returned if the command is currently invalid.
- 5 - Not a command - returned if the command is not a Mand2000 command.
- 6 - Failure - returned if a load or save operation fails.
- 7 - Out of range number - you know.
- 8 - Missing value.
- 9 - Miscellaneous - who knows?
- 10 - Syntax error - returned if you have passed a set of arguments that do not match the syntax string for that command.
- 25 - Unrecognized command.

## 1.5 printing\_from\_arexx

### Debugging Mand2000 ARexx scripts

If you run Mand2000 from a shell, then any ARexx commands that you run from Mand2000 can use the 'say' command to print information to that shell. If you run Mand2000 from the workbench however, the 'say' command may not produce any results, because there is nowhere for it to print to. To fix this you need to select the icon, select 'Information' from the workbench menu, and make sure that the 'window=con:...' tool type has no parentheses around it. If the tooltype is there and not commented out with parentheses, then a console window will open when needed by the say command. This is very important for debugging of ARexx scripts.

## 1.6 startupscript

### Startup Script

---



When Mand2000 starts up, it attempts to run an ARexx script called `'rexx:Startup.mnd2'`. This script is useful for customizing your Mand2000 environment in any way you want. The startup script which is supplied with Mand2000 adds a number of menus to the 'User' menu, allowing easy access to many of the other scripts.

## 1.7 portnames

ARexx port names:

Each fractal window in Mand2000 has its own ARexx port, allowing you to individually address them easily. The first window opened will have the name `MAND2000.1`, and the number increments as subsequent windows are opened. Although there is no limit on the number of fractal windows you can have open, there is a limit on the number of ARexx ports that a single copy of Mand2000 can have. Any windows after the thirteenth will not have ARexx ports - I hope that that is not too severe a limitation. To send a command to the first window you would write a small program like this:

```
/* Send to first Mand2000 window. */  
  
address MAND2000.1  
  
REQUESTNOTIFY '"Hello? Is anyone there?'"
```

This should bring up a requester with a hello message in it. One problem with this script is that it can only send commands to window one. In order to make it send commands to window two, you would have to change it. Luckily most of the time the address command is not necessary. This is because most ARexx scripts will be executed from within Mand2000, and then ARexx automatically knows which port to send the commands to. In this way if a script is executed from window two, all of the commands will be sent to window two.

When writing scripts that will be executed from Mand2000, it is best to end their file names with `'.mnd2'`. If you do this and save them in the `REXX:` directory, the ARexx will be able to find them with even if only the file name is given.

In addition to the ARexx ports that every window has, Mand2000 creates an additional ARexx port, one which isn't attached to any of the windows. This ARexx port is used to get global information about Mand2000, such as how many windows are open and what their port names are. The name for this port is `MAND2000`. If multiple copies of Mand2000 are run, subsequent copies will have the name `MAND2000.B`, `MAND2000.C`, etc.

## 1.8 commands

---

---

Global commands - send to "MAND2000" port.

Help  
Help on global ARexx commands

GetAttr  
Get Global attributes

Non-global commands - send to "MAND2000.?" port.

New commands for Mand2000 2.0:

OpenAnim  
Open an animation for writing

OpenScreenAnim  
Open an animation for writing the screen

WriteAnimFrame  
Write a frame to an open animation

CloseAnim  
Close an open animation file

MinimizeHammies  
Adjust the Minimize Hammies flag

Miscellaneous commands:

Help  
Help on non-global ARexx commands

Verbose  
Toggle verbose mode

Menu  
Add a menu item to the user menu

ClearMenus  
Clear all user menus

ClearNMenus  
Clear some user menus

EventAction  
Tie a script to an event

LockGui  
Lock the user out

UnlockGUI

---

---

Let the use back in

IncPause

Increase the pause level

DecPause

Decrease the pause level

NOP

Nothing

SetMorphFactor

Set the Mandelbrot morph ratio

SetExtraAcc

Scroll

Scroll the fractal

GetAttr

Get Project Attributes

Project commands:

New

Open a new Mandelbrot window

NewMand

Open a new Mandelbrot window

NewJulia

Open a new Julia window

Open

Load a file

Reset

Reset to the default coordinates

Save

Save to the current file

SaveAs

Save to a particular file

SaveScreenAs

Save the screen to a particular file

Close

Close the current fractal window

Print

Quit

Zoom commands:

---

---

ZoomInToPixel

ZoomOutToPixel

ZoomInToFloat

ZoomOutToFloat

SetLocation

SetJulia  
    Set the julia seed

SetFractalType

GotoGood  
    Go to one of the built in good spots

Undo

Redo

ForceRedraw

ForceRecalc  
    Setup commands:

CycleColours

OpenColours  
    Load a new palette

SetColour  
    Set a particular palette entry

SetColourMapping  
    Setup one of the built in colour mappings

SetColourMap  
    Set a custom colour mapping

SetScreen  
    Move Mand2000 to a specified screen

MaxIters  
    Set the maximum iterations per pixel

NumDrawPasses

TitleBar  
    Toggle the title bar

BackDrop

Workbench

---

---

Move Mand2000 to the workbench

FreeBuffers

Free all iteration buffers

Window management commands:

MandToFront

Move Mand2000's screen to the back

MandToBack

Move Mand2000's screen to the front

MoveWindow

SizeWindow

ChangeWindow

Simultaneously move and size the window

WindowToFront

WindowToBack

ActivateWindow

ZoomWindow

Toggle the 'zoom' state of the window

Requester commands:

RequestResponse

Bring up an OK/Cancel requester

RequestNotify

Bring up a Continue requester

RequestFile

Get a file name from the user

RequestSaveFile

Get a file name for saving from the user

RequestString

Get a line of text from the user

Requester

Invoke or revoke the Mand2000 requesters

DisplayMessage

Asynchronously display text

OkayN

---

Bring up a message requester with customizable buttons ↔

## 1.9 AppName/OpenAnim

OPENANIM FILENAME/A,

Starting with version 2.0, Mand2000 has built in support for creating animations. This is usually done through use of an ARexx script (the only alternative is creating morph animations through use of the built in morph anim requester). When you issue the OpenAnim command Mand2000 opens the specified file (deleting what was previously there) and prepares to write out a standard Anim-5 animation. The current width, height, depth and screen mode are stored and are used for the rest of the animation.

Note that only one animation can be open in Mand2000 at a time.

SEE ALSO

```
OpenScreenAnim
,
WriteAnimFrame
,
CloseAnim
```

## 1.10 AppName/OpenScreenAnim

OPENSREENANIM FILENAME/A,

This command is exactly like OpenAnim except that it prepares Mand2000 to create an animation involving the entire screen. After executing this command the WriteAnimFrame command will write out the entire screen to the animation file, instead of just the active window.

SEE ALSO

```
OpenAnim
,
WriteAnimFrame
,
CloseAnim
```

## 1.11 AppName/WriteAnimFrame

WRITEANIMFRAME

This command takes no parameters. It writes out the current window (if an OpenAnim was done on this window) or the entire screen (if an OpenScreenAnim was done on this window) to the animation

---

file. If no animation file is open, this command fails.

SEE ALSO

```
OpenAnim
,
OpenScreenAnim
,
CloseAnim
```

## 1.12 AppName/CloseAnim

CLOSEANIM

This command takes no parameters. It closes the open animation file, if there is one. This command is safe to use when there is no animation file open. Animation files are automatically closed by Mand2000 when:

The screen closes for any reason.

The window the animation was attached to is closed or sized.

Mand2000 exits.

Important note: This command will close ANY open animation file, even if the current window did not open it.

SEE ALSO

```
OpenAnim
,
OpenScreenAnim
,
WriteAnimFrame
```

## 1.13 AppName/MinimizeHammies

MINIMIZEHAMMIES ON/S,OFF/S

MINIMIZEHAMMIES is used to set the value of the Minimize Hammies flag (see the documentation of the 'Set misc' requester). If no parameters are specified then the state of the flag is unchanged. The current state of the flag is always returned in result as a one or a zero.

## 1.14 AppName/Help (Global)

HELP COMMAND,PROMPT/S

Help on its own returns a list of all of the global commands in the result string. If a command name is given, then the syntax string for that command is returned. If the PROMPT keyword is present, then the 'Help & Info' requester is brought up. In addition to getting

---

help through the help command, if you pass a command to Mand2000 with a question mark after it, Mand2000 will return the syntax string in the result. The global help command is identical to the project help command. The only difference is that the two commands are sent to different ARexx ports. The global ARexx port has a much smaller command set.

Examples:

```
HELP
HELP GETATTR
HELP PROMPT
```

See the PrintCommands.mnd2 script for example code.

SEE ALSO

Help (project)

## 1.15 AppName/GETATTR (Global)

GETATTR FIELD STEM/K

This commands fills in a stem variable with the global settings. If no stem variable is specified, the result is put in MAND2000ATTR. The 'Field' option is currently ignored. The values put into the stem variable are:

```
Stem.VERSION = version number of Mand2000.
Stem.COUNT = number of fractal windows.
Stem.1 = port name for the first fractal window.
Stem.2 = port name for the second fractal window...
Stem.PRIORITY = programs priority.
Stem.PALETTE = RGB values of the entire palette, printed
as R1-G1-B1 R2-G2-B2... All RGB values range from 0 to 255, even on non-AGA
machines.
Stem.ZOOMSPEED = zoom speed from the set misc requester.
Stem.WORKBENCH = one if on the workbench, otherwise zero.
Stem.SCREENDEPTH = number of bitplanes.
Stem.SCREENWIDTH = width of screen.
Stem.SCREENHEIGHT = height of screen.
Stem.DISPLAYID = actual screen mode number.
```

Examples:

```
GETATTR STEM STATVAR
say STATVAR.count
```

See the PrintAttr.mnd2 script for example code.

SEE ALSO

GetAttr (project)

## 1.16 AppName/Help (Project)

---



## HELP COMMAND, PROMPT/S

Help on its own returns a list of all of the global commands in the result string. If a command name is given, then the syntax string for that command is returned. If the PROMPT keyword is present, then the 'Help & Info' requester is brought up. In addition to getting help through the help command, if you pass a command to Mand2000 with a question mark after it, Mand2000 will return the syntax string in the result.

### Examples:

```
HELP
HELP GETATTR
HELP PROMPT
```

See the PrintCommands.mnd2 script for example code.

### SEE ALSO

Help (global)

## 1.17 AppName/Verbose

### VERBOSE ON/S, OFF/S

Verbose is used to turn the verbose flag on and off. When verbose is on, Mand2000 prints ARexx error messages. When verbose is off it is silent. If the ON or OFF switches are specified the flag is turned on or off. If neither switch is specified then the flag is unchanged. The behaviour is undefined if both switches are specified. The verbose flag is global. The current value of the verbose flag is always returned by this command.

### Examples:

```
VERBOSE ON
```

## 1.18 AppName/Menu

### MENU LABEL/A, SHORTCUT/K, CMD/F

The menu command is used to add a menu to the user menu. The label parameter specifies the text to appear in the menu, the shortcut parameter specifies an optional keyboard shortcut, and the remainder of the line is used as the ARexx command to be executed. The command should be the name of an ARexx script plus any needed parameters. The user menus are global. Note that in order to specify a keyboard shortcut you must use the shortcut keyword. Also note that if the shortcut is already used then the previous use will take precedence. The numbers make good shortcuts because none of them are used by Mand2000. Careful quoting must be used to preserve spaces and case in your parameters. New menu items are always added at the bottom of the list.

Examples:

```
MENU ' "Start Iter Movie"' itermovie start
MENU ' "Set Max Iters"' Shortcut X setmaxiters
```

See the Startup.mnd2 script for example code.

SEE ALSO

```
ClearMenus
,
ClearNMenus
,
EventAction
```

## 1.19 AppName/ClearMenus

CLEARMENUS

This command removes all of the user created menus, created with the MENU command.

SEE ALSO

```
Menu
,
ClearNMenus
,
EventAction
```

## 1.20 AppName/ClearNMenus

CLEARNMENUS

This command removes a specified number of user created menus, starting from the bottom of the list.

See the IterMovie.mnd2 script for example code.

SEE ALSO

```
Menu
,
ClearMenus
,
EventAction
```

## 1.21 AppName/EventAction

---

## EVENTACTION NAME/A,CMD/F

This very powerful command lets you tell Mand2000 to call an ARexx scripts whenever a particular action happens. This allows you to customize the operation of Mand2000 considerably. The most common use of this would be to set up a script that is called whenever a picture finishes calculating. This script could save the picture and then zoom, scroll, or load in another picture to be calculated. The name parameter specifies the name of the event and the CMD parameter specifies the ARexx command to be executed, which should be the name of an ARexx script plus any needed parameters. All of the event names below should be self explanatory.

PictureStarted  
PictureDone  
MovieStarted  
MovieDone  
MorphMovieStarted  
MorphMovieDone  
MouseClicked  
MouseDoubleClicked  
F1KeyPress  
F2KeyPress  
F3KeyPress  
F4KeyPress  
F5KeyPress  
F6KeyPress  
F7KeyPress  
F8KeyPress  
F9KeyPress  
F10KeyPress  
ScreenChange  
ColourMapChange  
LocationChange

SEE ALSO

Menu  
,  
ClearMenus  
,  
ClearNMenus

## 1.22 AppName/LockGui

## LOCKGUI

This command disables all of the Mand2000 menus, and tells Mand2000 to ignore all keystrokes and gadget presses. This is useful if you want to ensure that there is no interference with an ARexx script. **WARNING:** After a lockgui command is execute, the user can not exit Mand2000 until you execute an unlockgui. Don't leave the user with no way out.

SEE ALSO

---

UnlockGUI

## 1.23 AppName/UnlockGUI

UNLOCKGUI

This command unlocks the graphical user interface.

SEE ALSO

LockGUI

## 1.24 AppName/IncPause

INCPAUSE

This command increments the pause level. When the pause level is greater than zero, the relevant window in Mand2000 will be paused.

When an ARexx script is called from Mand2000, Mand2000 continues running. The user can still zoom and scroll, and Mand2000 continues calculating. However, the pause flag on window that the ARexx script was executed from is incremented. This stops calculations in that window only, which generally makes the ARexx/Mand2000 interaction cleaner. If you want Mand2000 to resume calculating while your script is still executing, simply decrement the pause flag, and then increment it when you want it to stop calculating.

SEE ALSO

DecPause

,

Misc

## 1.25 AppName/DecPause

DECPAUSE

This command decrements the pause level. When it reaches zero, the relevant window in Mand2000 will resume calculating. The pause level never goes negative. When an ARexx script begins executing the pause level will usually be set to one. If the window was already paused, the pause level will be set to two. Mand2000 decrements the pause level whenever an ARexx script terminates, so if the pause level is one or zero when a script exits, Mand2000 will resume calculating.

SEE ALSO

IncPause

,

Misc

## 1.26 AppName/NOP

NOP

This command does nothing.

## 1.27 AppName/SetMorphFactor

SETMORPHFACTOR FACTOR/A

This command is used to set the morph factor, as seen in the 'Morph Movie' requester. The valid range is from 0.0 to 1.0. This command only applies to the Mandelbrot set.

## 1.28 AppName/SetExtraAcc

SETEXTRAACC EXTRAACC/N/A

This command is equivalent to adjusting the extra accuracy in the 'Set Max Iters' requester. The value range of EXTRAACC is from 0 to 8. This adjust how early Mand2000 moves to the next accuracy level. If extra accuracy is set to four, then Mand2000 will jump to the next higher level of accuracy four zooms earlier than normally. This is only necessary if some calculation errors are seen.

## 1.29 AppName/Scroll

SCROLL DELTAX/N,DELTAY/N

This command scrolls the fractal picture, in the same way as keyboard, mouse and joystick scrolling work. The valid range for the scroll distances is the width and height of the drawing area. Positive numbers cause a scroll down and to the left, negative numbers cause a scroll up and to the right.

## 1.30 AppName/GetAttr (Project)

GETATTR FIELD,STEM/K

This commands fills in a stem variable with the settings specific to this window. If no stem variable is specified, the result is put in MAND2000ATTR. The 'Field' option is currently ignored. The values put into the stem variable are:

---

Stem.TOP = variable precision coordinate of the windows top edge.  
Stem.RIGHT = variable precision coordinate of the windows right edge.  
Stem.BOTTOM = variable precision coordinate of the windows bottom edge.  
Stem.LEFT = variable precision coordinate of the windows left edge.  
Stem.CENTERX = variable precision coordinate of the windows x center.  
Stem.CENTERY = variable precision coordinate of the windows y center.  
Stem.JULIAX = variable precision coordinate of the windows julia seed x.  
Stem.JULIAY = variable precision coordinate of the windows julia seed y.  
Stem.MAXITERS = current maximum iterations.  
Stem.PAUSE = pause flag - non zero if paused.  
Stem.DONE = done flag - non zero if the picture is finished.  
Stem.PERMILLAGE = zero to one thousand, representing amount done.  
Stem.FRACTALTYPE = fractal type - currently from zero to five. Odd numbers are Julia sets.  
Stem.DRAWLEVEL = current draw level, from -1 to 3. Negative one means finished.  
Stem.BACKDROP = the backdrop flag for this window.  
Stem.WIDTH = the full width of this window.  
Stem.HEIGHT = the full height of this window.  
Stem.DRAWWIDTH = the drawing width of this window.  
Stem.DRAWHEIGHT = the drawing height of this window.  
Stem.NUMZOOMS = the number of two times zooms from the outer Mandelbrot.  
Stem.EXTRAACC = the number of bits of extra accuracy.  
Stem.PRECISION = the number of sixteen bit words precision. On machines with 32 bit multiply instructions and floating point units the actual precision ← used may be greater.  
Stem.AREXX = the AREXX port name for this window.  
Stem.MASTERAREXX = the global AREXX port for this copy of Mand2000.  
Stem.FILENAME = the full path name for the current window, if present.  
Stem.FILE = the file name only for the current window, if present.  
Stem.MORPHFACTOR = the current morph factor, from 0.0 to 1.0.  
Stem.COLOURMAPTYPE = the name of the current colour map type.  
Stem.COLOURMAPSKIP = the current colour mapping skip value.  
Stem.COLOURMAPOFFSET = the current colour mapping offset value.  
Stem.MOUSEX = the mouse x position relative to this windows drawing area.  
Stem.MOUSEY = the mouse y position relative to this windows drawing area.  
Stem.MORPH.MAKING = the making morph movie flag.  
Stem.MORPH.FRAMESLEFT = the remaining frames to calculate in the morph movie.  
Stem.MORPH.FRAMENUM = the current morph movie frame number.  
Stem.MORPH.BASENAME = the morph movie base name.  
Stem.MORPH.ENDFACTOR = the destination morph movie morph factor.  
Stem.MOVIE.MAKING = the making zoom movie flag.  
Stem.MOVIE.FRAMESLEFT = the remaining frames to calculate in the zoom movie.  
Stem.MOVIE.FRAMENUM = the current zoom movie frame number.  
Stem.MOVIE.BASENAME = the current zoom movie base name.  
Stem.MOVIE.TWEENING = the current zoom movie tweening flag.  
Stem.MOVIE.ZOOMX = variable precision coordinate of the zoom movie x destination.  
Stem.MOVIE.ZOOMY = variable precision coordinate of the zoom movie y destination.  
Stem.MOVIE.ZOOMAMOUNT = the amount to zoom per frame in the zoom movie.

See the PrintAttr.mnd2 script for example code.

SEE ALSO

GetAttr (global)

### 1.31 AppName/New

```
NEW PORTNAME/K
NEWMAND PORTNAME/K
```

These two commands are identical. They both open up a new Mandelbrot window, exactly the same as selecting New Mandelbrot from the menus. If you specify portname, then that name will be used for the ARexx port name instead of the default one. This command returns the port name of the new fractal window in the result variable, useful if you want to send commands to the new window.

SEE ALSO

```
NewJulia
,
Close
```

### 1.32 AppName/NewJulia

```
NEWJULIA PORTNAME/K
```

This command opens up a new Julia window, exactly the same as selecting New Julia from the menus. If you specify portname, then that name will be used for the ARexx port name instead of the default one. This command returns the port name of the new fractal window in the result variable, useful if you want to send commands to the new window.

SEE ALSO

```
NewMand,
Close
```

### 1.33 AppName/Open

```
OPEN FILENAME/K
```

This command loads the specified file. If the load fails then the return code is set to 3. If the load succeeds then result will contain a number specifying what was loaded. The following values are added together to create the result:

```
1 - Location information loaded
2 - graphics loaded
4 - iterations loaded
8 - palette loaded
```

## 1.34 AppName/Reset

RESET

This performs the same action as selecting Reset from the menus. The coordinates of the current window are set to the 'factory defaults' of +/-2 by +/- 2.

## 1.35 AppName/Save

SAVE

This saves the file under its current name. If no name is set then the file requester is brought up.

SEE ALSO

SaveAs  
,  
SaveScreenAs

## 1.36 AppName/SaveAs

SAVEAS NAME/K,DATATYPE/N

This saves the file under the specified name, or brings up the file requester if the name is not specified. The datatype field specifies what data is saved. The location is always saved. The legal values for datatype are zero to three as follows:

- 0 - Save location only.
- 1 - Save location and graphics.
- 2 - Save location and iterations.
- 3 - Save location and iterations and graphics.

If the save fails then the return code is set to 6.

SEE ALSO

Save  
,  
SaveScreenAs

## 1.37 AppName/SaveScreenAs

SAVESCREENAS NAME/K

This saves the screen under the specified name, or brings up the file requester if the name is not specified. If the save fails then

---



the return code is set to 6.

SEE ALSO

```
Save
,
SaveAs
```

### 1.38 AppName/Close

CLOSE FORCE/S

This command closes the window that the message is sent to. If there is only one window left then the 'force' switch disables the quit confirmation request. If there is more than one window left, then the 'force' switch has no effect.

SEE ALSO

```
NewJulia
, NewMand,
Quit
```

### 1.39 AppName/Print

PRINT PROMPT/S

Print by itself prints with the current settings. If the 'prompt' switch is specified then print requester appears. There is currently no way to control the print settings from ARexx.

### 1.40 AppName/Quit

QUIT FORCE/S

Quit exits the program. If the 'force' switch is specified then Mand2000 will quit without asking for confirmation from the user.

SEE ALSO

```
Close
```

### 1.41 AppName/ZoomInToPixel

ZOOMINTOPIXEL X/N, Y/N, AMOUNT

This command zooms Mand2000 in towards a specific pixel

---

location. If either of the x or y coordinates is left out, then that coordinate defaults to the center of the window. 'Amount' is the amount that the magnification should be increased in the zoom. If no zoom amount is specified then the magnification is increased by 2.0, and an animated zoom is performed. If a zoom amount is specified then the picture begins redrawing at the higher magnification, centered around the specified location. The valid range for AMOUNT is from 1.0 to 100.0.

Examples:

```
ZOOMINTOPIXEL 10 10 1.10
```

SEE ALSO

```
ZoomInToFloat
,
ZoomOutToPixel
,
ZoomOutToFloat
```

## 1.42 AppName/ZoomOutToPixel

```
ZOOMOUTTOPIXEL X/N, Y/N, AMOUNT
```

This command is exactly like the ZOOMINTOPIXEL command except that a zoom out is performed instead of a zoom in.

Examples:

```
ZOOMOUTTOPIXEL 10 10 1.10
```

SEE ALSO

```
ZoomInToPixel
,
ZoomInToFloat
,
ZoomOutToFloat
```

## 1.43 AppName/ZoomInToFloat

```
ZOOMINTOFLOAT X, Y, AMOUNT
```

This command is exactly like the ZOOMINTOPIXEL command except that the location to zoom in towards is specified in terms of the fractal coordinates, instead of the pixel coordinates. To get the fractal coordinates for a point, either look at the set location requester, the status window, or the CENTERX and CENTERY fields after a GETATTR command.

Examples:

```
ZOOMINTOFLOAT -0.776608 0.134611 1.10
```

SEE ALSO

```
ZoomInToPixel
',
ZoomOutToPixel
',
ZoomOutToFloat
```

## 1.44 AppName/ZoomOutToFloat

ZOOMOUTTOFLOAT X,Y,AMOUNT

This command is exactly like the ZOOMINTOFLOAT command except that a zoom out is performed instead of a zoom in.

Examples:

```
ZOOMINTOFLOAT -0.776608 0.134611 1.10
```

SEE ALSO

```
ZoomInToPixel
',
ZoomInToFloat
',
ZoomOutToPixel
```

## 1.45 AppName/SetLocation

SETLOCATION TOP/A,RIGHT/A,BOTTOM/A,LEFT/A,JULIAX,JULIAY

This command allows you to numerically specify the coordinates of a picture. The four coordinates you must specify are the same numbers that show up in the set location requester. The JULIAX and JULIAY parameters are the optional julia seed coordinates. These are ignored if the fractal is not a Julia fractal. If one of JULIAX and JULIAY is specified, then the other must be specified also. The left value must be smaller than the right value and the top value must be larger than the bottom value. If there are errors in any of the numbers, the operation is cancelled.

SEE ALSO

```
SetJulia
```

## 1.46 AppName/SetJulia

---

SETJULIA JULIAX, JULIAY

This command allows you to numerically specify the julia seed coordinates, without changing the fractal location coordinates.

SEE ALSO

SetLocation

## 1.47 AppName/SetFractalType

SETFRACTALTYPE TYPE/A, JULIA/S

This command allows you to set what fractal type is used. The three types that Mand2000 supports are Mandelbrot, Mandcubed and Mandfourth. If your machine does not have an FPU (a 68881, 68882 or 68040) then only the Mandelbrot fractal is available. The fractal type is specified by name. Each fractal has an associated Julia set - you can request the associated Julia set by specifying the Julia parameter. This command is equivalent to changing the fractal type in the 'set location' requester.

Examples:

```
SETFRACTALTYPE Mandelbrot Julia
SETFRACTALTYPE Mandcubed
```

## 1.48 AppName/GotoGood

GOTOGOOD SPOTNUM/N/A

This command is equivalent to selecting a good spot from the list in the menus. The spotnum argument is the number of the good spot, with the first one being 1. There are currently seventeen good spots.

## 1.49 AppName/Undo

UNDO

This command is equivalent to the undo command in the menus.

SEE ALSO

Redo

---

## 1.50 AppName/Redo

REDO

This command is equivalent to the redo command in the menus.

SEE ALSO

Undo

## 1.51 AppName/ForceRedraw

FORCEREDRAW

This command forces a redraw. The redraw will happen with the previously calculated data, if it is available. This command should only be needed if the graphics data somehow gets corrupted.

SEE ALSO

ForceRecalc

## 1.52 AppName/ForceRecalc

FORCERECALC

This command forces Mand2000 to recalculate the current picture as well as redraw it. It is equivalent to selecting 'Force recalc' from the menus.

SEE ALSO

ForceRedraw

## 1.53 AppName/CycleColours

CYCLECOLOURS ON/S, FORWARDS/S, WAIT/N, SKIP/N

This command adjust the colour cycling. If the ON flag is specified, colour cycling starts, otherwise it stops. If FORWARDS is specified the cycling goes forwards, otherwise it goes backwards. WAIT is the number of 'vertical blanks' that happen between colour changes. Vertical blanks typically happen between fifty and seventy two times per second. SKIP is the number of colour registers that the colour cycling jumps on each cycle. WAIT must be from one to twenty, SKIP must be from one to four. Colour cycling does not work on the workbench screen.

---

## 1.54 AppName/OpenColours

OPENCOLORS FILENAME/K

This command is equivalent to the 'Open Colours' command in the palette requester menus. It returns eight if the load succeeded, zero otherwise. Colours can not be loaded on the workbench screen.

SEE ALSO

SetColour

## 1.55 AppName/SetColour

SETCOLOUR COLOURNUM/N/A,R/N/A,G/N/A,B/N/A

This command sets the RGB values of a single colour register. The register number and the Red, Green and Blue values must be specified. All four parameters must be in the 0 to 255 range. Colours can not be changed on the workbench screen.

SEE ALSO

OpenColours

## 1.56 AppName/SetColourMapping

SETCOLOURMAPPING TYPE,SKIP/N,OFFSET/N

This command controls the colour mapping. Currently six values are accepted for TYPE. These are the same values as are displayed in the type gadget in the 'Colour Mapping' requester. Skip and offset correspond to the skip and offset values in the colour mapping requester. The legal range for skip is from one to thirty two. The legal range for offset depends on the number of colours in your screen, but is the same as the range in the requester. The skip and offset values do not affect all colour mapping types. The valid types are:

REPEATED  
FRONTSPREAD  
SPREAD  
BACKSPREAD  
MONO  
FILLEDMONO

Examples:

SETCOLOURMAPPING repeated 3 10

SEE ALSO

SetColourMap

---

## 1.57 AppName/SetColourMap

SETCOLOURMAP MANDCOLOUR/N,MAPARRAY/F

The SetColourMap command is useful for those cases where the built in colour mapping options don't offer enough flexibility. This command allows you to specify the colour mapping for each colour individually, thus allowing you complete flexibility. The MANDCOLOUR parameter is the colour register number (from zero to max colour) that you would like used for all of the areas in the Mandelbrot set, the areas that are normally black. The MAPARRAY argument is a list of numbers, typically one for each iteration level, although fewer can be specified. The first number is used for those pixels that have an iteration count of zero, the second number is used for those with an iteration count of one, etc. Although this routine can be just called directly, if you do that then the next time that the number of iterations changes (every time you zoom) the old colour map will reappear. Typically this routine will be used with the 'EVENTACTION ColourMapChange' command, so that it gets called every time a new colour map is requested.

Examples:

```
SETCOLOURMAP 1 0 2 0 3 0 4 0 5 0 6 0 7 /* Stripes. */  
SETCOLOURMAP 0 /* White Mandelbrot. */
```

See the CustomColourMap.mnd2 script for example code.

SEE ALSO

SetColourMapping

## 1.58 AppName/SetScreen

SETSCREEN WIDTH/N,HEIGHT/N,DEPTH/N,DISPLAYID/N

This command tells Mand2000 to close its current screen and open a new one with the specified values. The DISPLAYID specifies the type of screen and can be obtained with the global GETATTR command. If any of the values are omitted then the screen mode requester appears.

SEE ALSO

Workbench

## 1.59 AppName/MaxIters

MAXITERS ITERCOUNT/N/A

This command sets the maximum number of iterations that will be performed per pixel, exactly the same as the 'Set Max Iters' requester does. The picture will be recalculated and redrawn as necessary. If the picture has already been calculated at a higher iteration level and

if there was enough memory to buffer all of the iterations, then no recalculation will be required and the redraw will be extremely quick.

## 1.60 AppName/NumDrawPasses

NUMDRAWPASSES NUMPASSES/N

This command sets the number of draw passes. The valid range for the parameter is from one to four. This setting will have no effect until the next time that drawing starts.

## 1.61 AppName/TitleBar

TITLEBAR ON/S,OFF/S

TITLEBAR is used to make the title bar visible or invisible. This command is only valid if Mand2000 is on a custom screen and there is a backdrop window, because the title bar can only become invisible if there is a backdrop window for it to hide behind. If no parameters are specified then the state of the title bar is unchanged. The current state of the title bar is always returned in result as a one or a zero.

## 1.62 AppName/BackDrop

BACKDROP ON/S,OFF/S

BACKDROP is used to change whether the window is a backdrop window or not. This command is only valid if Mand2000 is on a custom screen. If no parameters are specified then the backdrop state is unchanged. The current status is always returned in result as a one or a zero.

## 1.63 AppName/Workbench

WORKBENCH ON/S,OFF/S

WORKBENCH is used to change whether Mand2000 is on the workbench screen or not. If no parameters are specified then the workbench state is unchanged. The current status is always returned in result as a one or a zero.

SEE ALSO

SetScreen

---



## 1.64 AppName/FreeBuffers

FREEBUFFERS

This command is equivalent to the 'Free Buffers' command in the menus. It frees up the iteration buffers for all fractal windows. This can reduce Mand2000's memory usage substantially, but it means that if the colour mapping or number of iterations is changed that the picture will have to be recalculated.

## 1.65 AppName/MandToFront

MANDTOFRONT

This brings whatever screen that Mand2000 is on to the front.

SEE ALSO

MandToBack

## 1.66 AppName/MandToBack

MANDTOBACK

This pushes whatever screen that Mand2000 is on to the back.

SEE ALSO

MandToFront

## 1.67 AppName/MoveWindow

MOVEWINDOW LEFTEDGE/N, TOPEdge/N

This command moves the window to the specified location. If either of the values are invalid then no action will be performed and an error code will be returned.

SEE ALSO

SizeWindow

,

ChangeWindow

---

## 1.68 AppName/SizeWindow

SIZEWINDOW WIDTH/N,HEIGHT/N

This command sizes the window to the specified size. If either of the values are invalid then no action will be performed and an error code will be returned. Currently the minimum size is 53 pixels by 53 pixels. It is worth nothing that this command does work on backdrop windows, if that is what you want. If you shrink a backdrop window then you can also move it.

SEE ALSO

MoveWindow  
,  
ChangeWindow

## 1.69 AppName/ChangeWindow

CHANGEWINDOW LEFTEDGE/N,TOPEdge/N,WIDTh/N,HEIGHt/N

This command simultaneously sizes and moves the window to the specified size and location. If any of the values are invalid then no action will be performed and an error code will be returned. See SIZEWINDOW for more information.

SEE ALSO

SizeWindow  
,  
MoveWindow

## 1.70 AppName/WindowToFront

WINDOWTOFRONT

This command brings the window to the front of the screen, if possible.

SEE ALSO

WindowToBack

## 1.71 AppName/WindowToBack

WINDOWTOBACK

This command pushes the window to the back of the screen, if possible.

---

SEE ALSO

WindowToFront

## 1.72 AppName/ActivateWindow

ACTIVATEWINDOW

This command attempts to activate the window.

## 1.73 AppName/ZoomWindow

ZOOMWINDOW

This command is equivalent to selecting the zoom gadget of the window - it changes it to its alternate size.

## 1.74 AppName/RequestResponse

REQUESTRESPONSE PROMPT/F

This command brings up a yes/no style requester and waits for the user to respond to it. If the user selects 'Cancel' then RC is set to three, otherwise it is set to zero. The ARexx script will pause until this command returns. The vertical bar character '|' can be used to specify new lines.

SEE ALSO

```
RequestNotify
,
DisplayMessage
,
OkayN
```

## 1.75 AppName/RequestNotify

REQUESTNOTIFY PROMPT/F

This command brings up a text display requester and waits for the user to respond to it. The ARexx script will pause until this command returns. See DISPLAYMESSAGE if you want your script to continue running while the requester is up. The vertical bar character '|' can be used to specify new lines.

SEE ALSO

---

```
DisplayMessage  
,  
RequestResponse  
,  
OkayN
```

## 1.76 AppName/RequestFile

```
REQUESTFILE TITLE/K,PATH/K,FILE/K
```

This command brings up a file requester, with the specified window title, directory name and file name. If the user selects a file then the file name is returned in the result variable. If no file name is selected then RC is set to 3 and 'result' is undefined. All of the operands are optional.

SEE ALSO

```
RequestSaveFile
```

## 1.77 AppName/RequestSaveFile

```
REQUESTSAVEFILE TITLE/K,PATH/K,FILE/K
```

This command is exactly the same as the REQUESTFILE command except that a save requester is used. The colouring is different on this requester and files can not be selected by double clicking. In addition, if the user enters a file name that already exists, a requester warns them of this and gives them a chance to cancel. This command should be used in preference to the REQUESTFILE command whenever the file name will be used for saving.

SEE ALSO

```
RequestFile
```

## 1.78 AppName/RequestString

```
REQUESTSTRING TITLE/K,DEFAULT/K
```

This command gets a string of text from the user with the specified window title and default value. The maximum string length is currently 120 characters. If the user cancels this requester then RC is set to 6 and 'result' is undefined. All of the operands are optional.

---

## 1.79 AppName/Requester

REQUESTER NAME/A,ON/S,OFF/S

This command is used to control the various Mand2000 specific requesters. NAME is the name of the requester to be controlled. If ON or OFF is specified then the requester appears or disappears. If neither switch is specified then the requester state is not changed. The current status of the requester is always returned in 'result' as one or zero. The list of requesters currently controllable are:

MAXITERS  
SHOWLOC  
SETLOC  
PRINT  
STATUS  
JULIASEED  
MORPH  
MOVIE  
COLOURMAPPING

Examples:

```
REQUESTER MAXITERS ON
```

See the IterMovie.mnd2 script for example code.

## 1.80 AppName/DisplayMessage

DISPLAYMESSAGE OFF/S,X/N,Y/N,PROMPT/F

This command is similar to REQUESTNOTIFY in that it brings up a text display requester. The main difference between the two is that with DISPLAYMESSAGE, the ARexx script continues executing while the requester is up. At a later date you can either replace the old message with a new one, or kill the requester from your script. X and Y are optional parameters to specify the location of the requester - if either one is not specified then that coordinate defaults to being near the mouse. If you want to remove the requester, just specify the OFF switch. The requester always comes up inactive, so that it doesn't interfere with the user. As with the other text display requesters, the vertical bar character '|' should be used to specify new lines.

Examples:

```
DISPLAYMESSAGE 0 0 prompt "Hello there!"
DISPLAYMESSAGE 0 0 prompt "Hello again!"
DISPLAYMESSAGE off
```

SEE ALSO

```
RequestNotify
,
RequestResponse
```

```
,  
OkayN
```

## 1.81 AppName/OkayN

```
OKAYN TITLE/A,MESSAGE/A,GADGETS/A
```

This command is yet another way of bringing up a requester to display text to the user, and possibly get a response back. This method has two main differences from the other message commands. One is that it uses the Intuition EasyRequest command. This is useful if you want to use system standard requesters, and it also allows you to specify the title bar, the message text, and the text to appear in the gadgets. Text for multiple gadgets is specified by putting vertical bars ('|') between gadget texts. Multiple lines in the message text should have vertical bars between lines. The other difference is that this requester stops Mand2000 - no processing will happen until the user responds to this requester. If the user selects the rightmost gadget, which should be the cancel gadget, then this function returns 0 in RESULT. Any other gadgets return values starting from one and increasing from left to right.

Examples:

```
OKAYN ' "The title" ' ' "The message|text in three|lines" ' ' "Win|Lose|Draw" '  
OKAYN ' "Title" ' ' "Body text" ' ' "OK" '
```

SEE ALSO

```
RequestNotify  
,  
RequestResponse  
,  
DisplayMessage
```